Creating sentiment analysis algorithm for Romanian language

Andrei Pruteanu
andrei.pruteanu@gmail.com

Nicolae Marinică
nmarinica@gmail.com

Mihail Păduraru
mihail.paduraru@protonmail.com

PyData, Cluj-Napoca
February 25, 2020

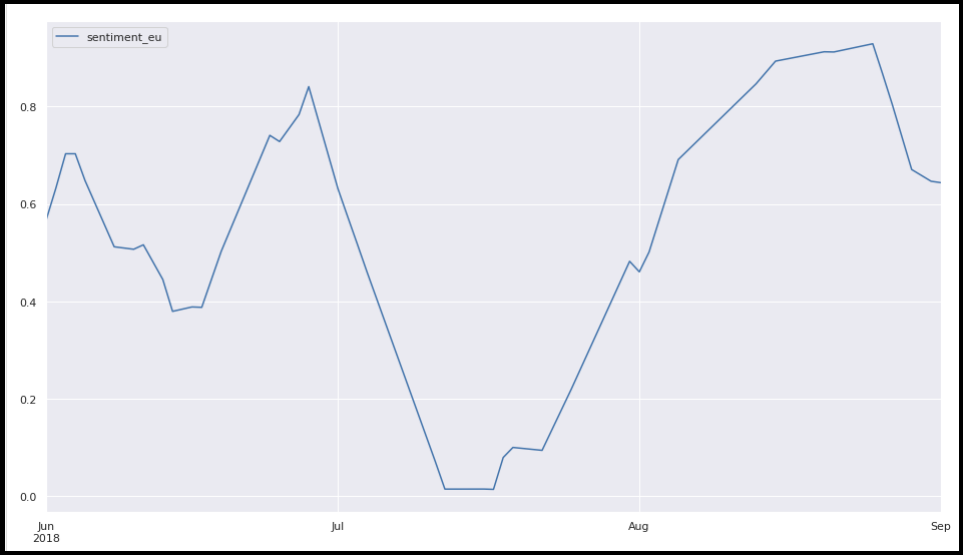
Outline

1. What is sentiment analysis
2. Data and cleaning
3. Sentiment analysis
4. Algorithm explainability
5. Word embeddings

What is sentiment analysis

Text-RO	Text-ENG	Sentiment
Filmul este fantastic! Imi place foarte mult!	The movie is fantastic! I like it a lot!	Positive
Filmul este so-so. As merge mai degraba la teatru.	The movie is so-so. I would rather go to the theater.	Neutral
Filmul este groaznic! Nu l-as viziona!	The movie is terrible! I would not watch it!	Negative

Historical media articles (TrustServista) - EU sentiment score trend



News data from: https://www.trustservista.com/

Training data

Data found here: https://github.com/katakonst/sentiment-analysis-tensorflow/

"Romanian dataset - Dataset of products and movies reviews"

Preprocess raw data

Convert all text to lower-case

Remove punctuation signs

Tokenization

Stemming

Balance the dataset

Data cleaning - stemming for Romanian language

word	stem	word	stem
abruptă	abrupt	ocol	ocol
absent	absent	ocolea	ocol
absentă	absent	ocolesc	ocol
absente	absent	ocoleşte	ocol
absenţa	absenţ	ocoleşti	ocol
absenţă	absenţ	ocoli	ocol
absenţi	absenţ	ocolim	ocol
absolut	absol	ocolind	ocol
absoluta	absol	ocolire	ocol
absolută	absol	ocolişuri	ocolişur
absolute	absol	ocolit	ocol
absolutul	absol	ocolită	ocol
absolutului	absol	ocoliţi	ocol

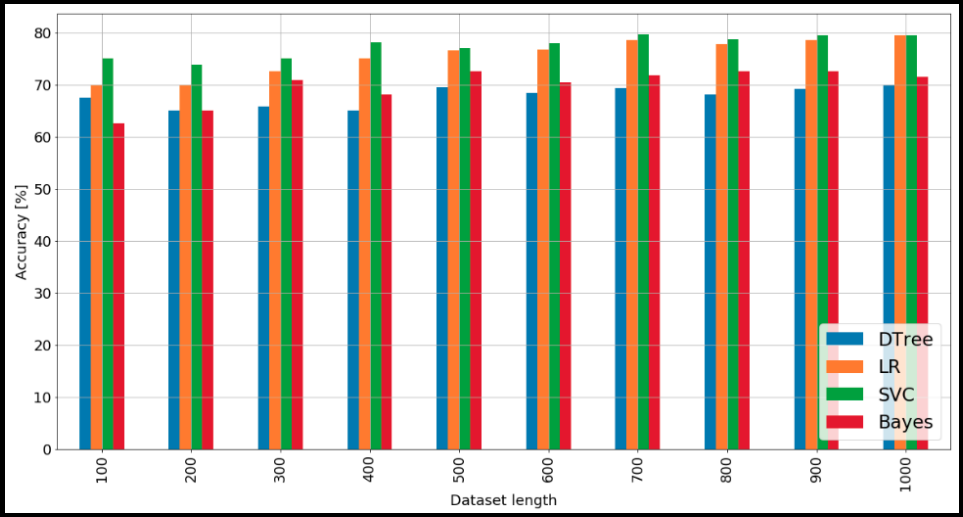
Preprocessing before vs after

	text	label
0	acest film a fost cel mai rau film pe care l-a...	0
1	calitatea de nostri a lui foley in acest film ...	1
3	creativitatea acestui film a fost pierduta de ...	0
4	cand am inchiriat acest lucru, speram ca ceea ...	0
5	acesta este un film de familie care incalzeste...	1
6	recomand\nraportul calitate-pret unul foarte bun	1
7	acest film parea promitator, dar de fapt era d...	0
8	este foarte amuzant. are o distributie mare, c...	1
9	am jurat mult timp in urma ca niciodata, sa ma...	0
10	unul dintre filmele cele mai nihiliste si brut...	1

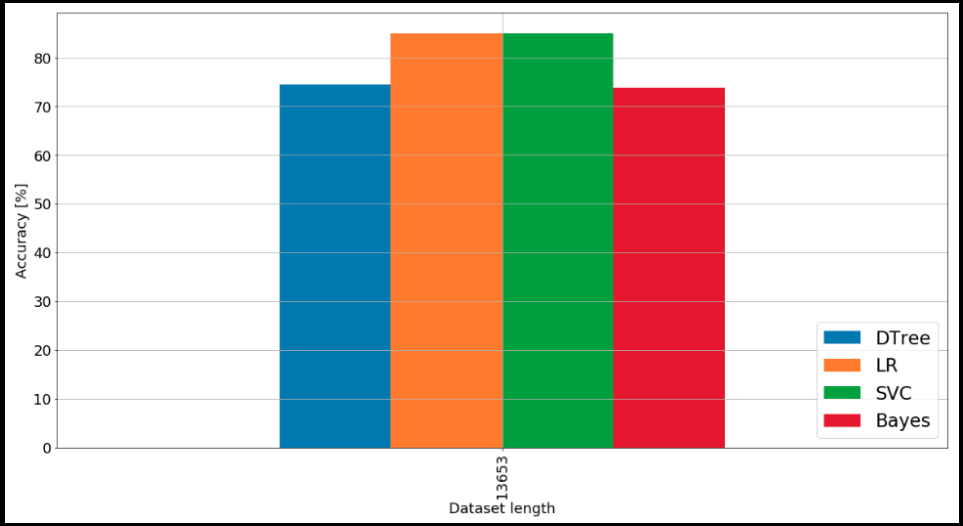
	text	label
0	[film, rau, film, l-am, vazut, vreodat, misiun...	0
1	[calitat, foley, film, satur, intens, celuloz,...	1
3	[creativ, film, pierdut, incep, scriitor, regi...	0
4	[inchir, lucru, sper, reign, of, fir, nast, ci...	0
5	[film, famil, incalzest, inim, absol, straluc,...	1
6	[recomand, raport, calitate-pret, bun]	1
7	[film, par, promit, fapt, dest, rau, premis, o...	0
8	[amuz, distribut, mar, perform, grozav, ales, ...	1
9	[jurat, timp, urma, niciod, uit, vreodat, film...	0
10	[film, nihilist, brutal, le-am, vazut, vreodat...	1

Demo

Classification algorithm compare - small datasets



Classification algorithm compare - complete dataset



SciKit regressor testing

```
1  test_sample1 = 'Filmul este fantastic! Imi place foarte mult!'
2  test_sample2 = 'Filmul este groaznic! Nu l-as viziona!'
3  test_sample3 = 'Filmul este so-so. As merge mai degraba la teatru.'
4
5  data = {'text': [test_sample1, test_sample2, test_sample3], 'label': [-1, -1, -1]}
6  data_df = pd.DataFrame.from_dict(data)
7
8  counts = count_vectorizer.transform(data_df['text'])
9  text = transformer.transform(counts)
10
11 models['lr'].predict(text)
```

```
array([1, 0, 0])
```

Model accuracy

	Category	Precision	Recall	F1
DTree	0	0.75	0.73	0.74
	1	0.74	0.76	0.75
LR	0	0.85	0.85	0.85
	1	0.86	0.85	0.85
SVC	0	0.86	0.84	0.85
	1	0.85	0.86	0.85
Bayes	0	0.75	0.70	0.73
	1	0.73	0.77	0.75

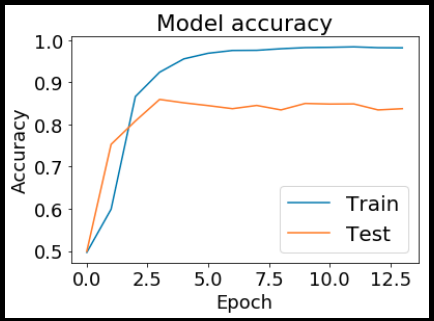
Model explainability

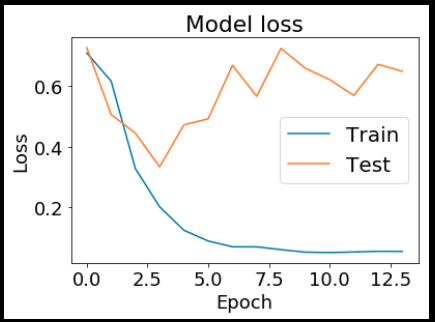
```
Document id: 11
Document text: astaz vazut film mi- placut lum creat pur simplu extraordinar imagin ved cinem 3d imax lucr m-a imp
resion povest ok general mi-au placut anum moment alte putin per total mi- placut film recomand vad lum minun par
real timp
True class: 1
Predicted class: 1
Explanations for class: 1
('placut', 0.01378709855431613)
('lum', 0.012687123392140088)
('minun', 0.012631616686001707)
('vazut', 0.010589176652991718)
('extraordinar', 0.008960931954320512)
('par', -0.00778420461172157)
('recomand', 0.007713920539033889)
('vad', 0.007595783852817189)
('timp', -0.006865655977861468)
('astaz', 0.006201770259938778)
```

Neural network model architecture

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 181, 100)          3006700
_________________________________________________________________
gru_1 (GRU)                  (None, 64)                31872
_________________________________________________________________
batch_normalization_1 (Batch (None, 64)                256
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 65
=================================================================
Total params: 3,038,893
Trainable params: 3,038,765
Non-trainable params: 128
_________________________________________________________________
None
```

Neural network model training





Neural network model testing

4.5. Test Model

```
1  test_sample1 = 'Filmul este fantastic! Imi place foarte mult!'
2  test_sample2 = 'Filmul este groaznic! Nu l-as viziona!'
3  test_sample3 = 'Filmul este so-so. As merge mai degraba la teatru.'
4  test_samples = [test_sample1, test_sample2, test_sample3]
5
6  test_samples_tokens = tokenizer_obj.texts_to_sequences(test_samples)
7  test_samples_tokens_pad = pad_sequences(test_samples_tokens, maxlen=max_length)
8
9  # predict
10 model.predict(x=test_samples_tokens_pad)
```

```
array([[0.9377453 ],
       [0.00610757],
       [0.06692801]], dtype=float32)
```

Word2Vec word similarity

```
1 gensim_model.wv.most_similar('politist')
```

```
[('interioar', 0.9962716102600098),
 ('betiv', 0.995508074760437),
 ('intalnest', 0.9940503239631653),
 ('portret', 0.9936536550521851),
 ('interconect', 0.9932307004928589),
 ('impotr', 0.9924734830856323),
 ('conflict', 0.9922173619270325),
 ('anchet', 0.9921332597732544),
 ('centrat', 0.992021918296814),
 ('leag', 0.991718053817749)]
```

Word2Vec odd word out

```
# odd word out
print(gensim_model.wv.doesnt_match("rebel impotr protagonist copac".split()))
```

```
copac
```

Thank you!